UNSTRUCTURED MESH GENERATION FOR VISCOUS FLOW COMPUTATIONS

Yasushi Ito[1] and Kazuhiro Nakahashi[2]

[1]*Tohoku University, Sendai, Japan, itoh@ad.mech.tohoku.ac.jp*
[2]*Tohoku University, Sendai, Japan, naka@ad.mech.tohoku.ac.jp*

ABSTRACT

This paper presents a robust and automated approach to generate unstructured hybrid grids comprised of tetrahedra, prisms, and pyramids for high Reynolds number flow computations. The hybrid mesh generation starts with isotropic tetrahedral grids to enhance the robustness. The tetrahedra near no-slip walls are shifted inward and the resulting gap between the tetrahedra and the walls is filled up with prismatic elements. The addition of prismatic elements is locally stopped if negative volume elements are created. The method is demonstrated for scaled experimental supersonic airplane models designed at the National Aerospace Laboratory of Japan (NAL) and an Aero-Train model designed at the Institute of Fluid Science, Tohoku University.

Keywords: unstructured hybrid meshing, engineering analysis, viscous flow analysis, CFD

1. INTRODUCTION

Computational Fluid Dynamics (CFD) has shown a prodigious growth in the last decades. To solve inviscid flows and low Reynolds number flows, generation of unstructured isotropic tetrahedral grids even around complex geometries has been demonstrated successfully. For high Reynolds number flows, however, such unstructured grids still have issues to be addressed. In order to accurately resolve thin boundary layers on the solid surface, a minimum grid spacing normal to the surface has to be very small. A solution is to create high aspect ratio elements near the surface, however it may cause a stiffness problem for the flow solver. Moreover, generation of such anisotropic stretched grids near the wall is another issue.

To treat the high Reynolds number flows by the unstructured grid, several methods of generating highly stretched grid on the surfaces have been proposed [1-10]. Some of them generate fully tetrahedral grids and the others create hybrid grids consisting of several elements such as prisms and tetrahedra. Although these methods have shown their capabilities in simulating viscous flows around three-dimensional configurations, their reliability still needs to be improved in order to treat various configurations encountered in engineering applications [11].

The authors proposed an automated unstructured hybrid mesh generation method based on isotropic tetrahedral grids [12]. The method started with isotropic tetrahedral grid generation for the entire computational domain. The tetrahedra near no-slip walls were shifted gradually to accommodate each prismatic layer. Prisms were then added near the walls keeping the mesh validity. The addition was locally stopped if negative volume elements were created, and pyramidal or tetrahedral elements filled the gap. By using the initial isotropic tetrahedral mesh as a background grid for the hybrid mesh generation, mesh quality control was improved greatly.

Another advantage of using isotropic tetrahedral meshes as the initial grid for the hybrid grid is that the grid validities can be checked by solving the Euler equations beforehand. The meshes are then used for generating hybrid meshes for viscous flow simulations. Required analysis time for a given configuration may be shortened.

The resulting method has shown its capability to generate hybrid meshes without user-interventions and the computed results have been validated against experiment. The mesh quality, however, still needs to be improved. For example, when prismatic layers were added to an isotropic tetrahedral mesh, only one normal vector was used at each point and skewed elements were created near the sharp corners. Special treatments for sharp ridges and corners, such as wing trailing edge and supersonic engine intake, are needed.

Sharov *et al.* [8] proposed a reliable viscous mesh generation method by modifying surface mesh generation to generate high-quality stretched tetrahedra near corners and ridges. Although its robustness is quite appealing, a large number of elements will be needed to represent complex geometries. Garimella *et al.* [6] employed multiple normal directions at sharp corners for creating anisotropic elements on no-slip walls. Although smoother transition between neighboring elements near the corners may be required, much less elements are needed. This idea will be implemented into the present approach to keep the required number of elements minimum.

In this paper, the improvement in the hybrid grid generation method based on the isotropic tetrahedral grid is presented. Hybrid mesh generation for several airplane models and the resulting flow computations is demonstrated.

2. SURFACE GRID GENERATION

Figure 1 shows the outline of the mesh generation process. CAD stereolithography (STL) files are employed for the surface definition. The STL file format is *de facto* standard in rapid prototyping, and the tessellated model defined by the STL file is precise enough to be used for CFD grid generation. The direct advancing front method based on geometric features is employed to generate surface grids [13]. This method is efficient and effective to generate fine surface meshes and have been demonstrated for various complex geometries successfully [14, 15].

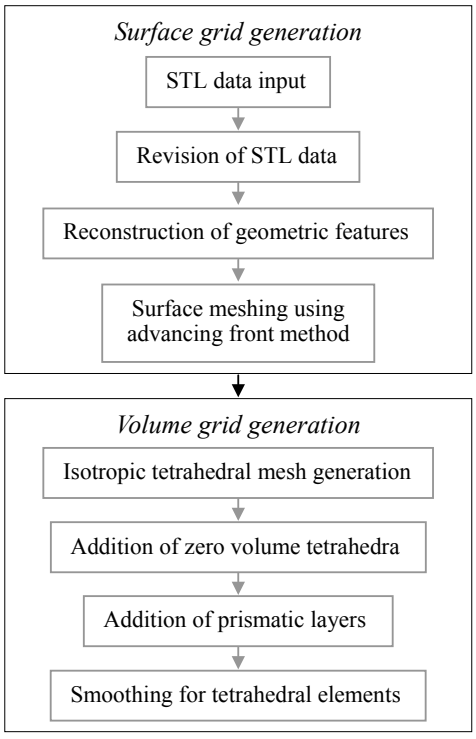


Fig. 1 Outline of grid generation process

3. VOLUME GRID GENERATION

3.1 Isotropic Tetrahedral Grid Generation

The entire computational domain defined by the surface mesh is first tessellated by isotropic tetrahedra [16]. Nowadays, techniques to generate isotropic tetrahedral grid has been well developed and any computational field about complex shapes can be fully automatically filled up with tetrahedral cells.

3.2 Addition of zero volume tetrahedra

An initial surface normal vector is calculated at each node on no-slip walls based on the visibility criterion [17]. This normal will be used as a marching direction for the prismatic layer generation except at nodes on sharp corners (defined as corner nodes). If a folding angle at each edge on no-slip walls is more than 150 degrees, the edge is considered as a part of a sharp corner. Two marching directions will be given to all corner nodes.

To employ the double marching directions, some pretreatments must be required because our hybrid mesh generation method starts with isotropic tetrahedral mesh. Therefore virtual elements—overlapping nodes and edges, zero area triangles, and zero volume tetrahedra—are added on sharp corners. Figure 2 shows how to modify a surface mesh at a sharp corner. In Fig. 2a, edge AB is on a sharp corner and nodes A and B are given double normals. In this case, nodes C and D, edges AC, BC, BD and CD, triangles ACB and BCD are added as shown in Fig. 2b. Nodes C and D have the same coordinates of A and B, respectively, accordingly lengths of edges AC and BD are zero, and triangles ACB and BCD have zero areas. Two zero volume tetrahedra, which have triangles ACB and BCD, respectively, are added. Each of the tetrahedra exists merely as a triangular face in this stage. The angles between the face and the surface normal vectors of the connecting corner nodes must be small in order to add prismatic layers easily without creating negative volume elements. The zero volume tetrahedra will have positive volumes when prisms are added.

The normals at nodes A, B, C and D are modified as follows:

$$\overline{\mathbf{N}}_X = \left| \overline{\mathbf{N}}_X^0 + \overline{\mathbf{N}}_{MX} \right| \ (X = A, B, C \text{ or } D) \qquad \overline{\mathbf{N}}_C^0 = \overline{\mathbf{N}}_A^0,\ \overline{\mathbf{N}}_D^0 = \overline{\mathbf{N}}_B^0 \tag{1}$$

where $\overline{\mathbf{N}}_X^0$ is the original surface normal, and $\overline{\mathbf{N}}_{MX}$ are calculated by use of an averaging of the face normals of the modified manifold as shown in Fig. 2b. If $\overline{\mathbf{N}}_X$ does not satisfy the visibility criterion, it should be corrected.

The calculated node normals are smoothed by the following simple Laplacian smoothing:

$$\overline{\mathbf{N}}_i^{n+1} = \left| \sum_j \overline{\mathbf{N}}_j^n \right| \tag{2}$$

where $\overline{\mathbf{N}}_i^n$ and $\overline{\mathbf{N}}_i^{n+1}$ are intermediate and modified node normals of node i, and j represents all surrounding nodes

belonging to the manifold of node i. This smoothing is not applied to corner nodes. The combination of the double normals at sharp corners and this smoothing method improve quality of prisms. To shorten required computation time, the smoothing starts from nodes near sharp corners. A final node normal $\overline{\mathbf{N}}_i$ is obtained from an initial node normal $\overline{\mathbf{N}}_i^0$ after the smoothing is applied for twenty times.

To correct height of prismatic layers in the following section, a correction value is prepared at all the nodes on no-slip walls.

$$c_i = \min\left(\frac{1}{\overline{\mathbf{N}}_i^0 \cdot \overline{\mathbf{N}}_i},\ 2 \right) \tag{3}$$

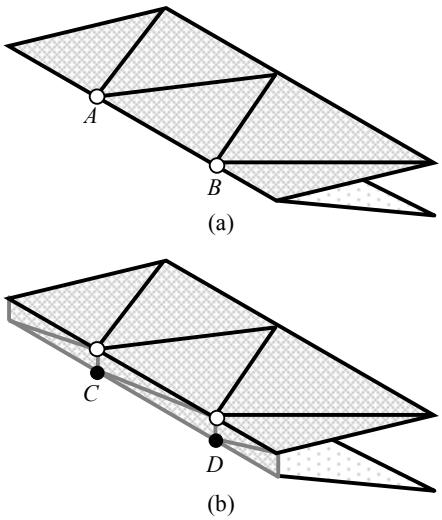


Fig. 2 Addition of elements at a sharp corner: (a) original surface mesh; (b) modified surface mesh

3.3 Generation of Prismatic Layers

User inputs to generate prismatic layers are only three parameters: the number of the layers n_p, an initial layer thickness near boundary walls $h_{\min}$, and a stretching factor f_s. No user intervention is required during the process. In this process, the addition of each prismatic layer is continued until the user-specified number of layers is obtained. Figure 3 illustrates the mesh generation process, which is summarized as follows:

1. Rank each node by the minimum number of edges connected to the no-slip wall. Let R_{Wi} be the rank value at node i. For example, nodes on no-slip walls are ranked to be zero, and nodes next to the walls are ranked to be one.
2. Rank each node by the minimum number of edges connected to an outer boundary or a "stagnation" node. Let R_{Oi} be this rank value at node i. Nodes on an outer boundary or stagnation nodes are ranked to be zero. The stagnation nodes are defined as follows:

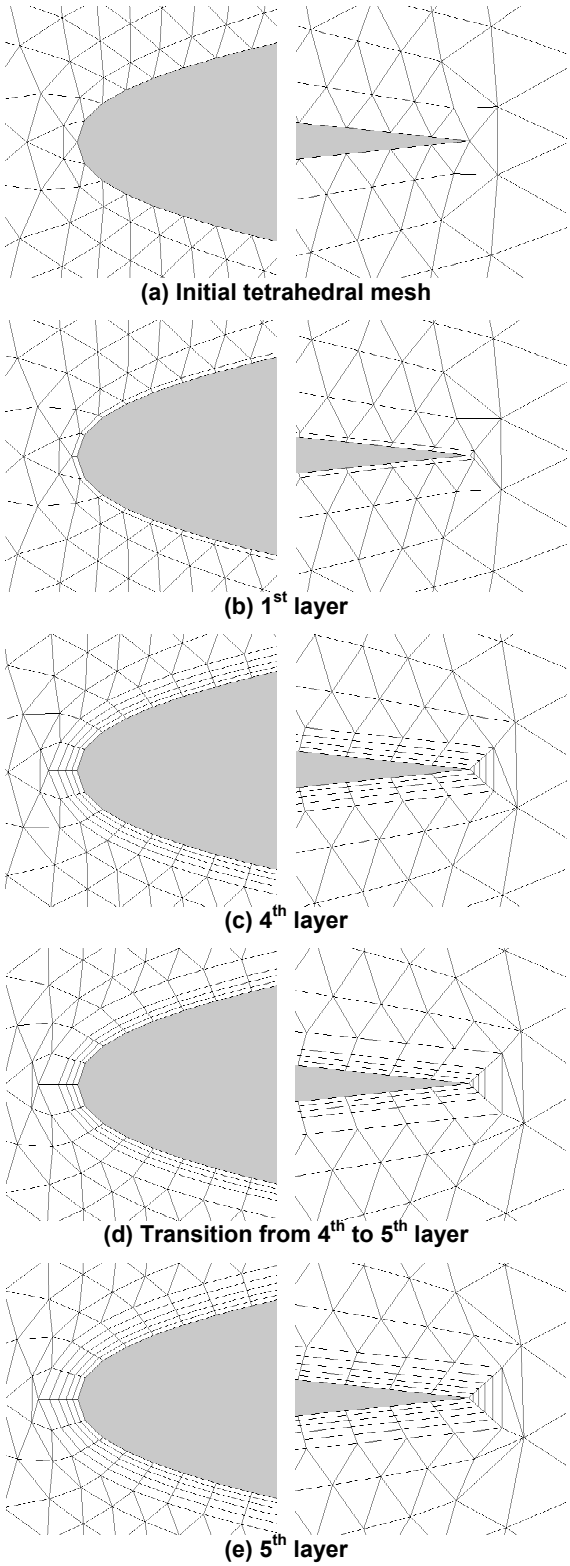


Fig. 3 Coarse mesh generation for an NACA 64A010 wing; its leading and trailing edge on the symmetry plane

i. Compare R_{Wi} with R_{Wj} at node i. Index j denotes all the neighboring nodes of node i. If $R_{Wj} > R_{Wi}$, node i is not a stagnation node.

ii. Shift no-slip walls by the total layer height hypothetically, and shift the inner nodes in accordance with the no-slip walls by a Laplacian-like method in order to check nodes' tendencies to move. The grid moving method to be mentioned in Step 7 may be diverted for this purpose. Let $\mathbf{H}_i$ be the resulting moving vector at node i, then calculate the following angle:

$$\theta_{ij} = \cos^{-1}\left(\overline{\mathbf{H}}_i \cdot \overline{\mathbf{H}}_j\right) \tag{4}$$

If the angle $\theta_{ij} > 70$ degrees, node i is considered as a stagnation node. The stagnation nodes may appear around the center of a tube and small gaps.

3. No-slip walls are initially considered as a front-line, and all the nodes expect those on outer boundaries are considered as movable. The surface normal at each node on no-slip walls, $\overline{\mathbf{N}}_i$, corresponds to a marching direction for adding prismatic layers.

4. The m-th prismatic layer (m = 1 to n_p) will be created under the front-line, and its height h_m is as follows:

$$h_m = f_s^{m-1} h_{\min} \tag{5}$$

5. At each node, initialize a displacement vector ($\mathbf{D}_i^0 = \mathbf{0}$).

6. Assign the following displacement vector $\mathbf{D}_i^0$ for each movable node on the front-line.

$$\begin{aligned} \mathbf{D}_i^0 &= \varepsilon \mathbf{D}_i \\ \mathbf{D}_i &= c_i h_m \overline{\mathbf{N}}_i \end{aligned} \tag{6}$$

where ε is a coefficient. c_i is calculated in Eq. 3, and this value is essential to create prisms that have almost same height from the front-line. While each node on the front-line will be shifted by $\mathbf{D}_i$, $\mathbf{D}_i^0$ is assigned in order to shift interior nodes adequately. ε is initially set as 1.03, and can be locally assigned a higher number when shifting a node on no-slip walls is difficult.

Note that a displacement vector is first assigned at each node in Steps 5-8, and then the nodes will be shifted in Step 9 based on the assigned vectors.

7. Assign a displacement vector for each of interior grid points employing the following weighted Laplacian-like method based on the displacement vectors on the front-line (Eq. 6).

$$\begin{aligned} \mathbf{D}_i^{n+1} &= \mathbf{D}_i^n + c\left(\frac{1}{\sum_j r_j} \sum_j r_j \mathbf{D}_j^n - \mathbf{D}_i^n \right) \\ r_j &= r_{Wj} + r_{Oj} \\ r_{Wj} &= \frac{1}{R_{Wj}+1},\ r_{Oj} = \frac{1}{R_{Oj}+1} \end{aligned} \tag{7}$$

where $\mathbf{D}_i^n$ and $\mathbf{D}_i^{n+1}$ are intermediate and modified vectors at node i, respectively, and c is a coefficient for the convergence acceleration (c = 1.44 in this case). The weighting value, r_j, is important to diffuse the assigned movements from nodes on the front-line adequately (r_{Wj}) and to keep isotropic shapes of tetrahedra inside tubes and at small gaps (r_{Oj}).

The number of iterations of Eq. 7, n, is less than twenty if applied nodes are sorted by the value, R_{Wi}.

8. Assign the displacement vector $\mathbf{D}_i$ instead of $\mathbf{D}_i^0$ for each movable node on the front-line.

9. Shift each movable node based on the vector $\mathbf{D}_i$ (see Fig. 3d, for example). Not to create negative-volume and flat tetrahedra, new node position should be restricted as follows. Figure 4 shows a node and its surrounding triangles in the two-dimension case. If the node is moved within the shaded zone, negative and flat elements are not created. In three-dimension, the distance between a node and its surrounding triangular faces must be greater than $0.5h_m$.

10. Check the actual moving distance d_{0i} ($\le |\mathbf{D}_i|$ due to the above restriction) at each movable node on the front-line.

i. If $0 \le d_{0i} \le 0.7|\mathbf{D}_i|$, node i cannot be shifted and is labeled as immovable.

ii. If $d_{0i} > 0.7|\mathbf{D}_i|$, node i can be shifted to accommodate new elements. However, node i will be labeled as immovable after Step 13 if $0.7|\mathbf{D}_i| < d_{0i} \le 0.95|\mathbf{D}_i|$.

11. Generate new elements under the front-line temporary as shown in Fig. 3e. If a face on the front-line has

i. one movable node, a tetrahedron is created.

ii. two movable nodes, a pyramid is created.

iii. three movable nodes, a prism is created.

12. Not to create negative volume elements, a control volume is calculated at each movable node on the front-line by using projection planes. Three different projection planes are employed to avoid computational errors. If the three computed volumes V_{1i}, V_{2i} and V_{3i} at node i satisfy at least one of the following conditions, the control volume is considered negative.

$$\begin{cases} V_{1i} < 0,\ V_{2i} < 0,\ V_{3i} < 0 \\ \left|\dfrac{V_{2i} - V_{1i}}{V_{1i}}\right| > 10^{-4},\ \left|\dfrac{V_{3i} - V_{1i}}{V_{1i}}\right| > 10^{-4} \end{cases} \tag{8}$$

If negative control volumes are detected, the corresponding nodes are labeled as immovable. Then all the newly created elements in Step 11 are removed and go back to Step 11.

13. If the number of prismatic layer already created is greater than $2n_p/3$, compare height of a connecting tetrahedron, T_t, and prism, T_p, at each face on the front-line. If $T_t <$

$1.7T_p$, the addition of prisms will be locally stopped here, i.e. the three nodes of the face is labeled immovable.

14. Update the front-line and continue Steps 4-13 until the user-specified number of layers is created.

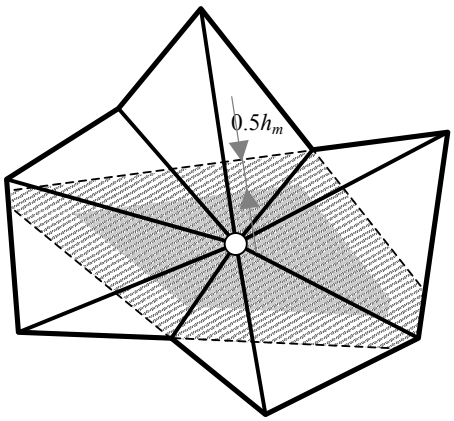


Fig. 4 New node location: the node can be moved within the shaded zone of its surrounding triangles in 2D.

3.4 Smoothing for Tetrahedral Elements

When prismatic layers are added, shapes of updated tetrahedra are not considered very much. After the prismatic layers are generated, the three smoothing methods are applied to the tetrahedral elements to improve the mesh quality: (1) edge collapse, (2) edge swapping and (3) Laplacian-like smoothing method. Two criteria are adopted whether those methods are applied or not: the volume ratio r_f at each triangular face and the ratio of the cube of the smallest edge of a tetrahedron to its volume r_t.

$$r_f = \frac{\min(V_A,\ V_B)}{\max(V_A,\ V_B)}$$
$$r_t = \frac{\min(V_A,\ v_A)}{\max(V_A,\ v_A)},\ v_A = d_{A\min}^3 \qquad (9)$$

where V_A is volume of tetrahedron A, and $d_{A\min}$ is the length of the smallest edge of tetrahedron A. Tetrahedra A and B share a triangular face.

3.4.1 Edge Collapse

If a tetrahedron has small r_t, the smallest edge is collapsed and the neighboring tetrahedra are removed.

3.4.2 Face Swapping

If two adjacent tetrahedra have small r_f, the jointing face may be swapped [18]. The configuration of the tetrahedra should be swappable and the resulting three tetrahedra must have higher quality.

3.4.3 Laplacian-like smoothing method

The following method is applied to the tetrahedral domain [19]:

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + c\left(\frac{1}{N}\sum_{j=1}^{N}\mathbf{x}_{oj}^n - \mathbf{x}_i^n\right) \qquad (10)$$

where $\mathbf{x}_i$ is a position vector of node i that is only shared by tetrahedral elements and is not adjacent to surface boundaries, c is the smoothing coefficient, N is the number of tetrahedral elements connecting node i, and $\mathbf{x}_{oj}$ is a centroid coordinate vector of one of the elements.

At each face, the volume ratio of neighboring tetrahedra r_f is computed. If the volume ratio r_f is smaller than 0.1, Eq. 10 is applied to the node that is shared by the smaller tetrahedron and that is not shared by the face. The smaller tetrahedra is marked and stored the value r_t to evaluate its quality. The smoothing coefficient c in Eq. 10 is initially set as 0.25. If there are negative volume elements or marked tetrahedra have smaller r_t when node i is moved, c is gradually reduced.

4. FLOW SOLVER

The full Reynolds-averaged Navier-Stokes equations that retain the unsteady form are solved by a finite volume cell-vertex scheme. The control volumes are non-overlapping dual cells constructed around each node. The Harten-Lax-van Leer-Einfeldt-Wada (HLLEW) Reimann solver [20] was used for the numerical flux computations. The Lower-Upper-Symmetric Gauss-Seidel (LU-SGS) implicit method [21] was used for time integration. A one-equation turbulence model by Goldberg and Ramakrishnan [22] was implemented to treat turbulent boundary layers.

5. APPLICATIONS

5.1 NAL NEXST-1 Airplane Model

The National Aerospace Laboratory of Japan (NAL) has been working on the scaled experimental supersonic airplanes [23, 24]. The project will develop two types of experimental airplanes: a rocket-launched unpowered airplane (NEXST-1 project, and a jet-engine-powered airplane (NEXST-2 project). In this section, the hybrid mesh generation applied to an NAL NEXST-1 airplane model is shown.

Figure 5 shows a generated hybrid mesh for the semi-span model, which contains 1,455,202 nodes, 1,880,730 tetrahedra 2,157,682 prisms and 62,827 pyramids. Parameters for adding the prismatic layers were as follows: the number of layers of 30, the initial layer thickness of $0.05/\sqrt{\mathrm{Re}}$ and the stretching factor of 1.2. At the sharp corners such as the wing trailing edge, double normal vectors were employed to enhance the mesh quality.

The Navier-Stokes equations were solved at a free-stream Mach number of 2.0, a unit Reynolds number based on meter of 27.5×10^6 and an angle of attack of 2.0°. Figure 6 shows

computed surface pressure distributions. Figure 7 shows the chordwise pressure coefficient (C_p) distribution at 30% semi-span stations. The wind tunnel data measured by Aerodynamic Design Group, Next Generation Supersonic Transport Project Center, NAL were plotted for comparison. At the wing trailing edge, the C_p distribution seems to be correct owing to employing double normal directions for adding prismatic layers. The C_p distribution at the same section for another hybrid mesh using only one normal direction is plotted in Fig. 8. Unphysical pressure jump is shown at the wing trailing edge due to the skewed prisms there.

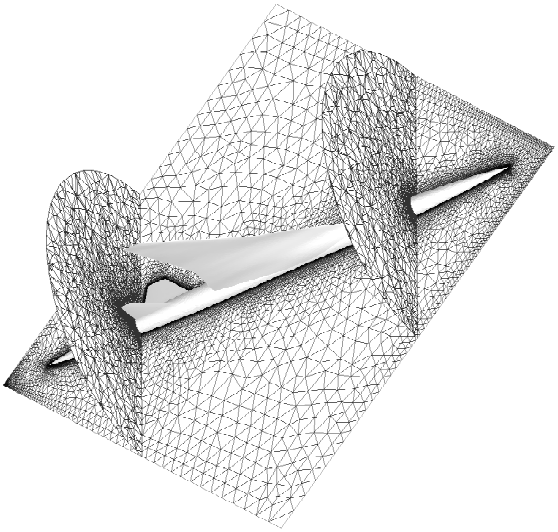


Fig. 5 Hybrid mesh for NEXST-1: (a) geometry; (b) wing leading edge at 30% semi-span station; (d) wing trailing edge at 30% semi-span station

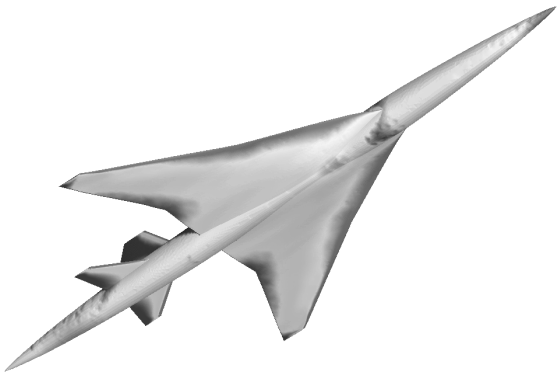
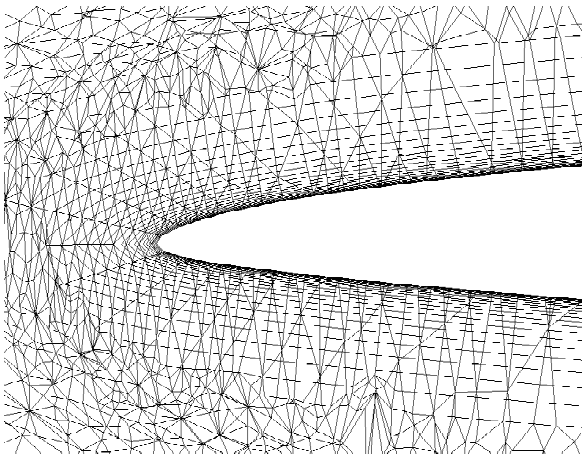
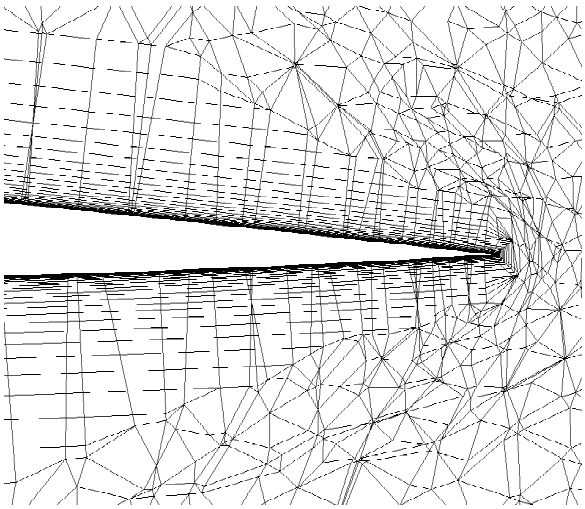


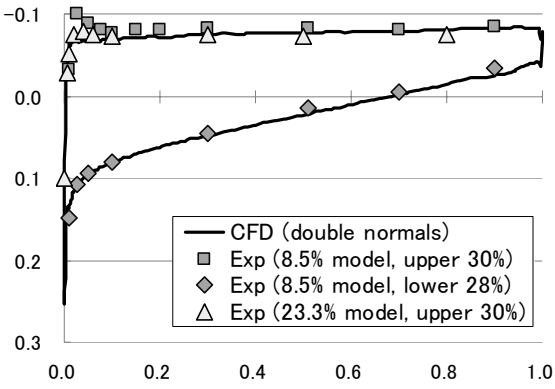
Fig. 6 Surface pressure distribution for NEXST-1



(a)

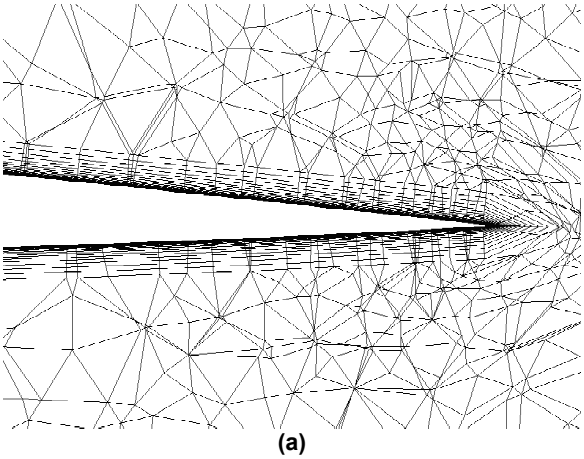


(b)

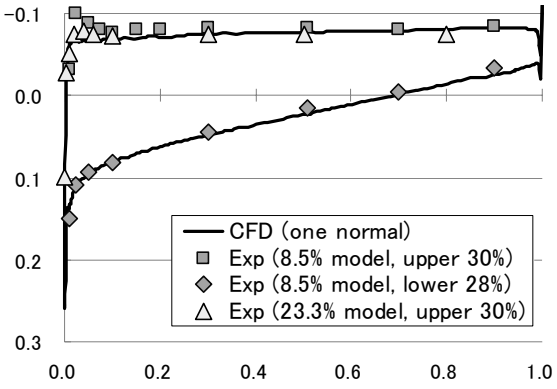


(c)

Fig. 7 Hybrid mesh for NEXST-1 at 30% semi-span station: (a) and (b) hybrid mesh around the leading and trailing edge employing double normals at sharp corners for adding prismatic layers; (c) C_p distribution



(a)

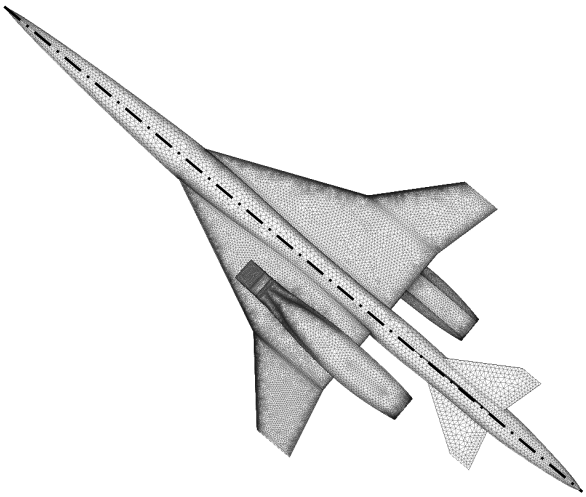


(b)

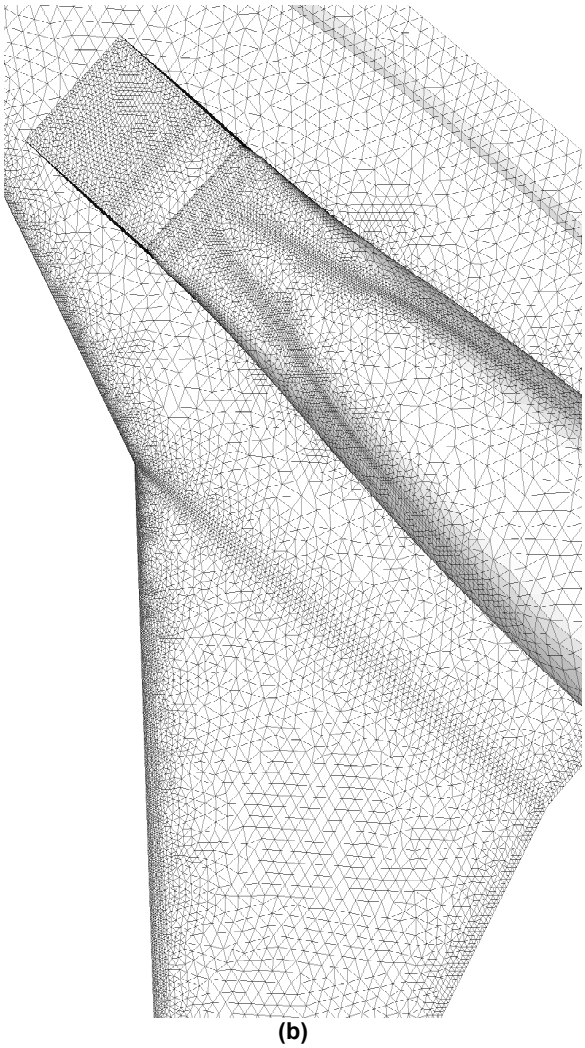
Fig. 8 Hybrid mesh for NEXST-1 at 30% semi-span station: (a) hybrid mesh around the trailing edge employing one normal at each node for adding prismatic layers; (b) C_p distribution

5.2 NAL NEXST-2 Airplane Model

In this section, the hybrid mesh generation was applied to an NAL NEXST-2 semi-span model having a flow-through engine nacelle. Its geometry is completely redesigned from NEXST-1. Figures 9 and 10 show its surface mesh and volume mesh of the semi-span model for the flow solver based on the Navier-Stokes equations. Smooth transition from prismatic to tetrahedral elements is shown. Parameters for adding the prismatic layers were as follows: the number of layers of 26, the initial layer thickness of $0.05/\sqrt{\mathrm{Re}}$ and the stretching factor of 1.25. The hybrid mesh consisted of 1,479,944 nodes, 2,030,116 tetrahedra, 2,165,432 prisms, 43,670 pyramids, 142,686 boundary triangles, and 12,300 boundary quadrangles. Figure 11 shows a surface pressure distribution and pressure contours on the symmetry plane at a free-stream Mach number of 1.7, a unit Reynolds number based on meter of 25.974×10^6 and an angle of attack of 2.0°.

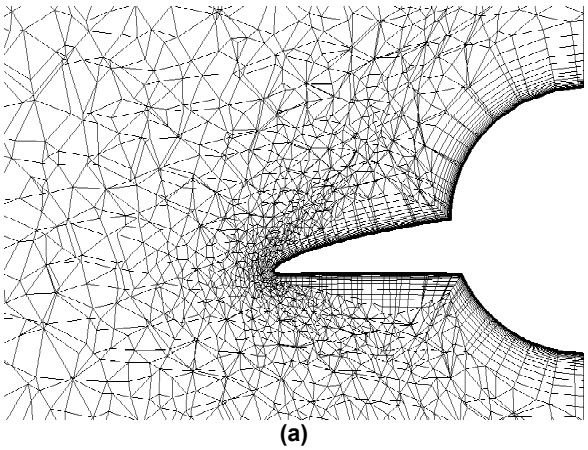


(a)

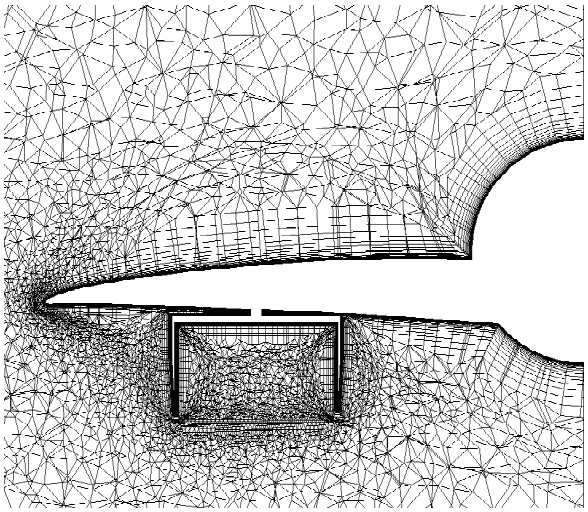


(b)

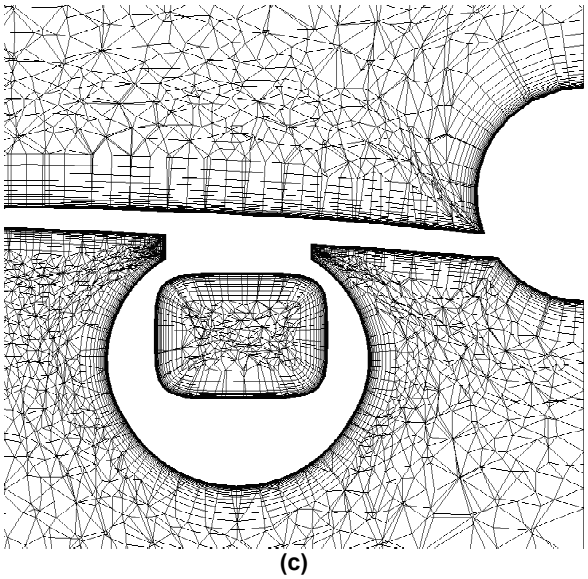
Fig. 9 Surface grid for NAXST-2: (a) general view; (b) close-up at the engine intake



(a)



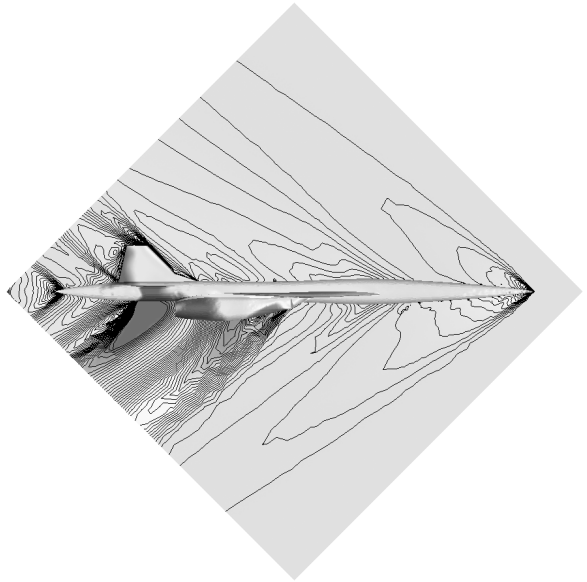
(b)



(c)

Fig. 10 Hybrid grid for NAXST-2: (a) cross-flow cross section at $x = 0.4L$; (b) $x = 0.5L$; (c) $x = 0.6L$

(a)



(b)

Fig. 11 Pressure distribution for NAL NEXST-2 (M_∞ = 1.7, $\overline{\mathrm{Re}} = 25.974 \times 10^6$, α = 2.0°): (a) surface pressure distribution; (b) pressure contours on the symmetry plane

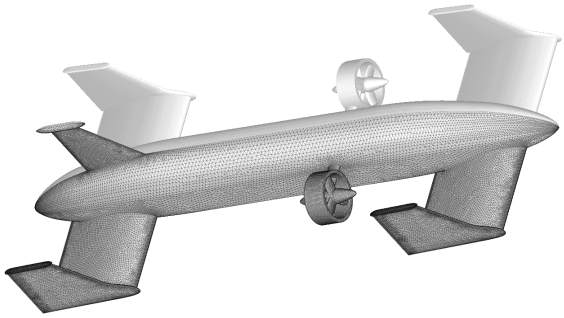


Fig. 12 Surface mesh for Aero-Train

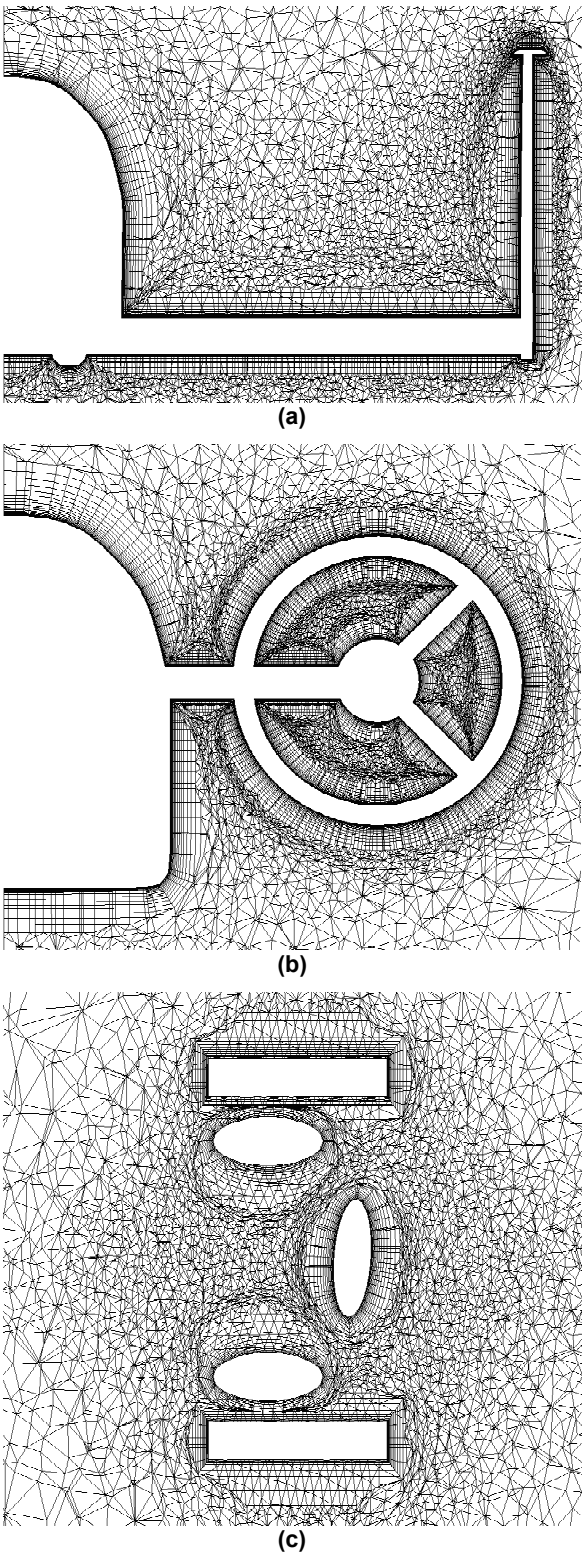


Fig. 13 Hybrid mesh for Aero-Train: (a) cross-flow cross section at x = 0.19L; (b) x = 0.39L; (c) spanwise cross section around the rotor at η = 0.65

5.3 Aero-Train

Figure 12 shows a surface mesh for a prototype of a high-speed vehicle called the Aero-Train currently under investigation at the Institute of Fluid Science, Tohoku Univ., Japan [25, 26]. The vehicle runs using the wing-in-ground effect. Figure 13 shows a hybrid mesh, which consisted of 1,322,749 nodes, 2,630,092 tetrahedra, 1,640,069 prisms and 48,321 pyramids. The Navier-Stokes equations were solved at a free-stream Mach number of 0.123, a Reynolds number of 23.6×10^6 and an angles of attack of 0°. Figure 14 shows the surface pressure distribution.

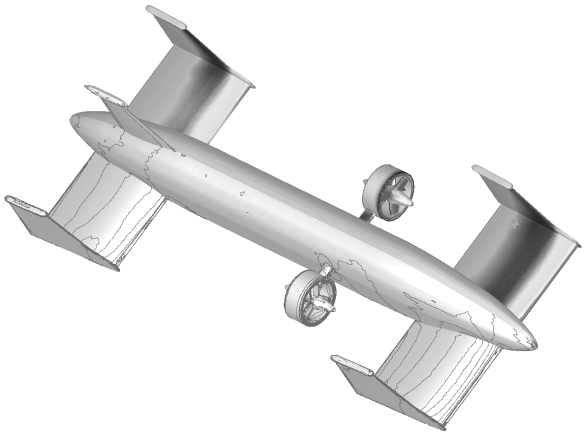


Fig. 14 Surface pressure distribution and contours for Aero-Train (M_∞ = 0.123, Re = 23.6 $\times10^6$, α = 0°)

6. CONCLUSIONS

An automated hybrid grid generation method based on isotropic tetrahedral meshes has been developed. The direct advancing front method was employed for the surface triangulation based on stereolithography data. For the robustness, the hybrid grid generation method started from isotropic tetrahedral grids that could be generated for any complex geometries. Each prismatic layer was then added step by step near no-slip walls automatically. Use of double normals improved grid quality around sharp corners. The method was applied to the NAL experimental airplane models and the Aero-Train. These confirmed that the proposed method practically generated well-qualified grid distribution for high Reynolds number flow computations.

ACKNOWLEDGEMENTS

The geometric data of the experimental supersonic airplanes and the wind tunnel data were provided by Aerodynamic Design Group, Next Generation Supersonic Transport Project Center, NAL. The geometric data of the Aero-Train were provided by Y. Kohama, a professor of the Institute of Fluid Science, Tohoku Univ. The authors would like to thank T. Fujita, a graduate student of Tohoku Univ., and T. Kato, a

lecturer of the Institute of Fluid Science, Tohoku Univ., for their help in generating the STL files. The Navier-Stokes equations were solved using the supercomputer in the Institute of Fluid Science, Tohoku Univ.

REFERENCES

[1] Marcum, D. L., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA Paper 95-0212, 1995.

[2] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43-49.

[3] Hassan, O., Morgan, K., Probert, E. J. and Peraire, J., "Unstructured Tetrahedral Mesh Generation for Three-Dimensional Viscous Flows," *International Journal for Numerical Methods in Engineering*, Vol. 39, 1996, pp. 549-567.

[4] Sharov, D. and Nakahashi, K., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flow Applications," *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 157-162.

[5] Khawaja, A. and Kallinderis, Y., "Hybrid Grid Generation for Turbomachinery and Aerospace Applications," *International Journal for Numerical Method in Engineering*, Vol. 49, 2000, pp. 145-166.

[6] Garimella, R. V., and Shephard, M. S., "Boundary Layer Mesh Generation for Viscous Flow Simulations," *International Journal for Numerical Method in Engineering*, Vol. 49, 2000, pp. 193-218.

[7] Michal, T., and Cary, A., "Unstructured Grid Extrusion for Viscous Flow Simulations," AIAA Paper 2001-0444, 2001.

[8] Sharov, D., Luo, H., Baum, J. D. and Löhner, R., "Unstructured Navier-Stokes Grid Generation at Corners and Ridges," AIAA Paper 2001-2600, 2001.

[9] Currie, T. C., "Modifying Isotropic Tetrahedral Grids for Viscous Simulations," Proceedings of the 7th Annual Conference of the Computational Fluid Dynamics Society of Canada, 1999.

[10] Löhner, R. and Cebral, J., "Generation of Non-Isotropic Unstructured Grids via Directional Enrichment," *International Journal for Numerical Method in Engineering*, Vol. 49, 2000, pp. 219-232.

[11] Wurtzler, K. E., Tomaro, R. F., and Witzeman, F. C., "Analysis of Three Unstructured Grid Techniques From a User's Perspective," Proceeding of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations, 2000, pp.357-366.

[12] Ito, Y. and Nakahashi, K., "Unstructured Hybrid Grid Generation Based on Isotropic Tetrahedral Grids," AIAA Paper 2002-0861, 2002.

[13] Ito, Y. and Nakahashi, K., "Direct Surface Triangulation Using Stereolithography Data," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 490-496.

[14] Fujita, T., Ito, Y., Nakahashi, K. and Iwamiya, T., "Computational Fluid Dynamics Evaluation of National Aerospace Laboratory Experimental Supersonic Airplane in Ascent," *Journal of Aircraft*, Vol. 39, No. 2, 2002, pp. 359-364.

[15] Ito, Y. and Nakahashi, K., "Surface Triangulation for Polygonal Models Based on CAD Data," *International Journal for Numerical Methods in Fluids*, Vol. 39, Issue 1, 2002, pp. 75-96.

[16] Sharov, D. and Nakahashi, K., "A Boundary Recovery Algorithm for Delaunay Tetrahedral Meshing," Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations, 1996, pp.229-238.

[17] Kallinderis, Y. and Ward, S., "Prismatic Grid Generation for Three-Dimensional Complex Geometries," *AIAA Journal*, Vol. 31, No. 10, 1993, pp. 1850-1856.

[18] Barth, T. J., "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations," in AGARD Report 787, Special Course on Unstructured Grid Methods for Advection Dominated Flows, 1992, pp. 6.1-6.61.

[19] Marcum, D. L. and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1619-1625.

[20] Obayashi, S. and Guruswamy, G. P., "Convergence Acceleration of an Aeroelastic Navier-Stokes Solver," *AIAA Journal*, Vol. 33, No. 6, 1995, pp. 1134-1141.

[21] Sharov, D. and Nakahashi, K., "Reordering of Hybrid Unstructured Grids for Lower-Upper Symmetric Gauss-Seidel Computations," *AIAA Journal*, Vol. 36, No. 3, pp. 484-486, 1998.

[22] Goldberg, U. C. and Ramakrishnan, S. V., "A Pointwise Version of Baldwin-Barth Turbulence Model," *Computational Fluid Dynamics*, Vol. 1, 1993, pp.321-338.

[23] Iwamiya, T., "NAL SST Project and Aerodynamic Design of Experimental Aircraft," Proceedings of the

4th ECCOMAS Computational Fluid Dynamics Conference, Athens, John Wiley & Sons Ltd., Chichester, UK, 1998, pp.580-585.

[24] Sakata, K., "Supersonic Experimental Airplane (NEXST) for Next Generation SST Technology," AIAA Paper 2002-0527, 2002.

[25] Kohama, Y., Watanabe, H., Kikuchi, S., Ohta, F., Itoh, T., "Flight Dynamics and Development of the Stability Control Method of the Aero-Train (1st Report, Flight Test by Pushing)," *Transactions of the Japan Society of Mechanical Engineers*, Vol. 68, No. 665, 2002, pp. 102-107 (in Japanese).

[26] Kohama, Y., "Aero-Train Project," http://www.ifs.tohoku.ac.jp/kohama-lab/getstope.html